**Klausur**

## Name, Vorname (Dozent/in) \_ Patrin, Alexander \_\_

## Name, Vorname (Teilnehmer/in) \_ Scurtu, Andrei \_

## Modul \_\_\_\_\_OCPJP\_(Java SE)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Zulässige Hilfsmittel: \_\_\_\_\_alle\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Bearbeitungszeit (Minuten)**  90\_\_\_\_\_

maximale Punktzahl 100\_\_\_\_

erreichte Punktzahl \_\_\_\_\_\_\_

**Note: \_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ Prozent: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

|  |  |
| --- | --- |
| **Note** | **Prozentbereich** |
| 1,0 | 100 - 95 |
| 1,3 | 94 - 89 |
| 1,7 | 88 - 85 |
| 2,0 | 84 - 81 |
| 2,3 | 80 - 77 |
| 2,7 | 76 - 73 |
| 3,0 | 72 - 69 |
| 3,3 | 68 - 65 |
| 3,7 | 64 - 60 |
| 4,0 | 59 - 55 |
| 4,3 | 54 – 50 |
| nicht bestanden | 49 – 0 |

Notenspiegel

Viel Erfolg!

27.11.2019

Ort, Datum Unterschrift (Teilnehmer/in)

Aufgabe 1. (2 Punkte)

Welche primitiven Datentypen werden von Java unterstützt? Richtige Schlüsselwörter ankreuzen.

[ + ] int

[ ] unsigned long

[ ] character

[ + ] byte

Aufgabe 2. (2 Punkte)

Wählen Sie die Vergleichsoperatoren aus.

[ ] ~

[ ] ?=

[ + ] >

[ + ] !=

Aufgabe 3. (2 Punkte)

Wählen Sie die arithmetischen Operatoren aus.

[ ] ^

[ + ] %

[ + ] \*

[ ] ?:

Aufgabe 4. (2 Punkte)

Kreuzen Sie die Java-Kontrollstrukturen an.

[ ] goto-Anweisung

[ ] run-Anweisung

[ + ] while-Schleife

[ + ] erweiterte for-Schleife

Aufgabe 5. (4 Punkte)

Welche Schleifen geben die Zahlen 1 bis 10 auf der Konsole aus?

[ + ] for(int i=1;i<=10;++i) { System.out.println(i); }

[ + ] for(int i=1;i<=10;i++) { System.out.println(i); }

[ ] for(int i=1;i<10;i++) { System.out.println(i); }

[ ] while(int i=1;i<10;i++) { System.out.println(i); }

Aufgabe 6. (4 Punkte)

Wie erzeugen Sie ein Array mit den Werten 1, 2, 3, 4, 5?

[ + ] int[] array = {1, 2, 3, 4, 5};

[ + ] int array[] = {1, 2, 3, 4, 5};

[ + ] int[] array = new int[] {1, 2, 3, 4, 5};

[ ] int[] array = new int[5] {1, 2, 3, 4, 5};

Aufgabe 7. (4 Punkte)

Gegeben ist ein int-Array "array". Wie benutzen Sie eine Schleife, um den Inhalt des Arrays auf der Konsole auszugeben?

[ + ] for(int i=0;i<array.length;i++) { System.out.println( array[i] ); }

[ ] for(int i=0;i<array.size;i++) { System.out.println( array[i] ); }

[ ] for(int i : array) { System.out.println(array[i]); }

[ + ] for(int i : array) { System.out.println(i); }

Aufgabe 8. (4 Punkte)

Was kompiliert?

[ + ] int x1 = 33;

[ ] int 1x = 33;

[ ] int x? = 33;

[ + ] int x$ = 33;

Aufgabe 9. (4 Punkte)

Gegeben:

for(int i=0;i<10;i++) {

if(i==4) continue;

System.out.print(i);

}

Welche Ausgabe erwarten Sie?

[ ] keine Ausgabe

[ ] 4

[ ] 0123456789

[ + ] 012356789

Aufgabe 10. (4 Punkte)

Gegeben:

for(int i=0;i<10;i++) {

if(i==4) break;

System.out.print(i);

}

Welche Ausgabe erwarten Sie?

[ ] 0123456789

[ ] 012356789

[ ] 01234

[ + ] 0123

Aufgabe 11. (4 Punkte)

Was trifft zu?

[ ] Objekte können auf dem STACK liegen

[ + ] Objekte können auf dem HEAP liegen

[ ] Die Instanzbildung ist stets mit dem Laden einer Klasse verbunden

[ + ] Die Instanzbildung ist stets mit dem Aufruf eines Konstruktors verbunden

Aufgabe 12. (4 Punkte)

Was trifft zu?

[ + ] Die Sichtbarkeit "public" gestattet den uneingeschränkten Zugriff aus allen Paketen

[ ] Elemente mit der Sichtbarkeit "protected" sind nur innerhalb der eigenen Klasse sichtbar

[ ] Die Sichtbarkeit "default" darf nur für die statischen Methoden eingesetzt werden

[ ] Nur abgeleitete Klassen können auf die "private" Elemente einer Klasse zugreifen

Aufgabe 13. (4 Punkte)

Gegeben ist:

class Foo {

Foo(int x) {

}

}

Was ist richtig?

[ + ] Die Klasse hat den default-Konstruktor

[ + ] Der explizite Konstruktor hat den default-Konstruktor deaktiviert

[ + ] Der explizite Konstruktor ruft super() auf

[ ] Der explizite Konstruktor ruft this() auf

Aufgabe 14. (4 Punkte)

Wie kann eine "IS-A" Beziehung definiert werden?

[ ] KlasseA implements KlasseB

[ + ] KlasseA extends KlasseB

[ + ] KlasseA implements InterfaceA

[ + ] InterfaceA extends InterfaceB

Aufgabe 15. (4 Punkte)

Was trifft zu?

[ + ] Von einer abstrakten Klasse können KEINE Objekte mit "new" erstellt werden

[ + ] Abstrakte Methoden sind in einer abstrakten Klasse erlaubt

[ + ] Von einer "final" Klasse kann man nicht erben

[ ] Es können KEINE Objekte von einer "final" Klasse erstellt werden

Aufgabe 16. (4 Punkte)

Gegeben:

String text="abcdef";

System.out.println(text.substring(3));

Welche Ausgabe erwarten Sie?

[ ] abc

[ ] abcdef

[ + ] def

[ ] cdef

Aufgabe 17. (4 Punkte)

Gegeben:

String text="abcdef";

System.out.println(text.substring(1,3).toUpperCase());

Welche Ausgabe erwarten Sie?

[ ] BCD

[ ] CD

[ + ] BC

[ ] AB

Aufgabe 18. (4 Punkte)

Gegeben:

String text="abcdef";

System.out.println(text.charAt(3));

Welche Ausgabe erwarten Sie?

[ ] b

[ ] c

[ + ] d

[ ] e

Aufgabe 19. (4 Punkte)

Gegeben:

String text="b";

System.out.println(text.compareTo("d"));

Welche Ausgabe erwarten Sie?

[ + ] -2

[ ] 0

[ ] +2

[ ] Keine Ausgabe

Aufgabe 20. (4 Punkte)

Gegeben:

String textA="Hallo";

String textB="Hallo";

System.out.println(textA==textB);

Welche Ausgabe erwarten Sie?

[ + ] true

[ ] false

[ ] Es gibt einen Compilerfehler

[ ] Es gibt eine Exception

Aufgabe 21. (4 Punkte)

Gegeben:

String textA=new String("Hallo");

String textB=new String("Hallo");

System.out.println(textA==textB);

Welche Ausgabe erwarten Sie?

[ ] true

[ + ] false

[ ] Es gibt einen Compilerfehler

[ ] Es gibt eine Exception

Aufgabe 22. (4 Punkte)

Was kompiliert?

[ + ] String s = (String) new Object();

[ ] String s = new Object();

[ + ] Object s = (Object) new String();

[ + ] String s = (String)(Object) new String();

Aufgabe 23. (4 Punkte)

Was verursacht eine Exception?

[ ] StringBuilder s = new Object();

[ + ] String s = (String) new Object();

[ ] StringBuilder s = (StringBuilder) new StringBuffer();

[ + ] Exception s = (Exception) new Throwable();

Aufgabe 24. (4 Punkte)

Was kompiliert?

[ + ] static void foo(int a, int... b) {}

[ ] static void foo(int... b, int a) {}

[ ] static void foo(int... a, int... b) {}

[ + ] static void foo(int[] a, int[] b) {}

Aufgabe 25. (4 Punkte)

Gegeben ist folgende Klasse:

public class Person {

private String vorname;

private String nachname;

int alter;

private boolean verheiratet;

public Person(String vorname, String nachname, int alter) {

this.vorname = vorname;

this.nachname = nachname;

this.alter = alter;

}

}

Wie wird die toString-Methode KORREKT (nicht unbedingt sinnvoll) überschrieben?

[ + ] public String toString() {

return vorname + " " + nachname + " ist " + alter;

}

[ ] String toString() {

return vorname + " " + nachname + " ist " + alter;

}

[ + ] public String toString() {

return "Person";

}

[ ] public String toString(String beruf) {

return vorname + " " + nachname + " ist " + beruf;

}

Aufgabe 26. (4 Punkte)

Gegeben ist die Klasse 'MyClass', die kompiliert:

public class MyClass {

static { OtherClass.doStuff(); }

public MyClass() {

OtherClass.doMoreStuff();

}

}

Und folgende main-Methode, die auch keine Compilerfehler hat:

public static void main(String[] args) {

new MyClass(); // Zeile A

}

Welche Exceptions oder Errors wird die Zeile A definitiv nicht werfen?

[ + ] java.lang.ExceptionInitializerError

[ + ] java.io.IOException

[ + ] java.lang.ArithmeticException

[ + ] java.lang.Exception

Aufgabe 27. (4 Punkte)

Was kompiliert?

[ + ] Predicate<String> p1 = s -> s != null;

[ ] Predicate<String> p2 = String s != null;

[ + ] Predicate<String> p3 = (String s) -> s != null;

[ + ] Predicate<String> p4 = (String s) -> { return s != null; };